Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Vivek Shah

Entitled EMBEDDING AND DETECTING MACHINE READABLE DATA IN IMAGES PRINTED BY ELECTROPHOTOGRAPHIC PRINTER

For the degree of Master of Science in Electrical and Computer Engineering

Is approved by the final examining committee:

J. P. Allebach
Chair

E. J. Delp

G. T. Chiu

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): J. P. Allebach

Approved by: M. J. T. Smith 4/23/08
Head of the Graduate Program Date

Graduate School Form 20
(Revised 10/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

EMBEDDING AND DETECTING MACHINE READABLE DATA IN IMAGES PRINTED BY ELECTROPHOTOGRAPHIC PRINTER

For the degree of Master of Science in Electrical and Computer Engineering

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Vivek Shah
Signature of Candidate

4/21/08
Date

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

EMBEDDING AND DETECTING MACHINE-READABLE DATA IN IMAGES PRINTED BY ELECTROPHOTOGRAPHIC PRINTER

A Thesis

Submitted to the Faculty

of

Purdue University

by

Vivek M. Shah

In Partial Fulfillment of the

Requirements for the Degree

of

Masters of Science in Electrical and Computer Engineering

May 2008

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

I would like to express deep appreciation to my advisor, Dr. Jan P. Allebach, for his guidance, encouragement and support through the course of my MS program. He provided me with the opportunity to work in the exciting field of image processing. His encouragement and patience have always helped me implement new ideas and overcome the obstacles that came in my way during this research experience. It has been an enriching experience to work with this exceptional researcher and a wonderful person. Without his guidance and persistent help this dissertation would not have been possible.

I would also like to thank the members of my graduate committee for their time and effort and for having faith in me.

I would like to thank the School of Electrical and Computer Engineering at Purdue and the National Science Foundation for the financial support of my work.

I would also like to thank all my colleagues in EISL labs for their love and support; especially Sungjoo Suh for always being there to help me through my research and Changyung Lee for his help in generating hybrid screen.

Lastly I would like to thank my parents and my friend, Avani Joshi, for believing in me and providing the moral support to go through this endeavor.

This material is based upon work partially supported by the National Science Foundation under Grant No. 0524540-CNS.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

SYMBOLS

dpi Dots per inch

ABBREVIATIONS

EP	Electrophotographic
HVS	Human Visual System
AM	Amplitude Modulation
FM	Frequency Modulation
DBS	Direct Binary Search
LSM	Least Squares Method
PWM	Pulse Width Modulation

ABSTRACT

Shah, Vivek M. M.S.E.C.E., Purdue University, May 2008. Embedding And Detecting Machine-Readable Data In Images Printed By Electrophotographic Printer. Major Professor: Jan P. Allebach.

Watermarking is a technique used to authenticate and distinguish original documents or images from the copied or forged ones. Many watermarking methods have been developed and are being used for printed documents and images. Though these current watermarking techniques can distinguish between original and forged documents, it is not possible to identify the source of the forged and counterfeit documents, knowledge of which is required in some cases to stop future forgeries. In this thesis we propose a machine readable watermarking technique for images, halftoned using clustered-dot halftoning technique and printed on electrophotographic printer, that can, not only identify forged documents and images, but also provide information about the printer used and date on which the document was printed.

First, we propose a method of embedding data in the image by shifting clustered-dot pair diagonally and horizontally. The range of gray level values within which data can be embedded, number of pixels by which dots should be shifted and location of shifted dots is decided based on experiments carried out.

Next, we propose a method for detecting data from the scanned image. Scanned image is preprocessed using various image processing algorithms and techniques before detection stage. The detection of embedded data is done from the knowledge of the centroids of each clustered-dot. Error control coding technique is also proposed to improve the accuracy of detection process.

Lastly we propose an embedding method using pulse width modulation to reduce the visual distortions in the image caused by embedding the data.

1. INTRODUCTION

1.1 Motivation

In recent years, there has been remarkable progress in the field of printing. The printers that printed documents at slow speed and rendered poor image quality have been replaced by fast working printers which render very good quality of printed document. Along with this, the cost of printers have also decreased significantly making them easily affordable. No longer is it necessary to have the high end expensive printers to obtain good quality of printed documents, as now it can be obtained from the low cost printers commonly found in most of the houses and offices.

During early days, forging documents and images was not that easy due to poor printing qualities by a low cost printer. It was essential to replace it by sophisticated equipments and skills to produce immaculate forgery. However, owing to the present scenario as mentioned above, counterfeiting of important documents and images like confidential papers, banknotes, passports, etc. has become a common act of fraud. Even a common desktop printer in the market works good enough to reproduce documents and images having quality as good as an original. This has increased the importance of research and study in the field of document security. Watermarking is one such technique that helps to identify between original and forged documents. But at the same time one would want to track the source of those forged documents. However, tracking the source of such forged documents and images has become difficult because these printers are available in most offices and households. Forged government documents, passports, etc. should be identified and prevented from use by wrong people. Counterfeit notes should be stopped from circulation and its source

should be identified to stop its production in the future. Authorities and federal agencies are encouraging research in the area that can help them detect and stop all these forgeries. It has become necessary to have digital watermark to all the original documents and images. But it should also be robust to noise and not detectable by people who want to forge documents. Therefore, a need arises for a development that embeds and detects machine readable information in the printed document for their verification.

1.2 Background

Continuous tone images are first transformed into halftoned image by a technique called Digital Halftoning [1] before printing them on any printer. Most printers are binary or multilevel with few levels. It is not possible to render a continuous tone image by limited number of levels in such printers. Digital halftoning is the method that allows the printers to render continuous tone images by two or more number of levels in the printers. It exploits the fact that human visual system (HVS) acts as a low pass filter [2] and so high frequency variations outside the cutoff frequency of the HVS is not perceived by humans. The main goals of halftoning are: 1) Representation of tone - Should produce smooth, homogeneous texture and free from visible structure of contouring; 2) Representation of detail - Should be free from moire effect and have detail rendition capability.

The halftone textures can be classified as clustered dot periodic or aperiodic and dispersed dot periodic or aperiodic. For our research, we will be considering clustered dot periodic halftoning textures. Based on modulation strategies halftoning is classified as Amplitude Modulation(AM) and Frequency Modulation(FM). In amplitude modulation the size of the dots is modulated with the gray level while in frequency modulation the frequency of dots is modulated with the gray level of the continuous tone image [1], [2]. As we are considering clustered dot periodic textures in our research, it can be classified as AM halftoning strategy.

Based on the computational complexity involved, halftoning algorithms can be classified into three main categories : point processes(screening or dithering), neighborhood processes(error diffusion), and iterative processes(Direct Binary Search (DBS)) [3]. Point processes do pixel by pixel comparison with spatially varying threshold to obtain the binary value of the halftone image. It requires fewer computations. Neighborhood processes calculate binary value at each pixel and also the error. It propagates the error to the neighboring pixels ahead of the current pixel in the scan direction. It requires more computation than point processes. Iterative processes process the image in multiple passes. The final halftone image is selected such that it minimizes the perceived error. They are computationally most complex.

Screening is a point-to-point process which transforms the continuous tone image to binary image without memory or information about the neighborhood. The halftoned image dot structure is based on the dot profile function of the screen. Dot profile function is the family of binary textures used to render each level of constant tone. There is a one-one relationship between dot profile and the screen. A M×N halftone cell has MN different dot size in dot profile function and the average absoprtance of those dots are 0, $\frac{1}{MN}$, ... , 1. The threshold values are selected based on the size of the halftone cell. The selection of threshold levels should be such that it is uniformly spaced over the range of gray values of the input image. In cluster dot textures, the thresholds that are close in value are located close together in the threshold matrix. As against that, in dispersed dot textures the thresholds close in value are located far apart. To achieve correct representation of tone, the size of the halftone cell should be as large as possible. But to achieve proper representation of detail, the size of halftone cell should be as small as possible. Thus, to achieve both of these goals, a number of halftone screens are grouped together as a single screen called macroscreen. A macroscreen is designed using search based optimization strategies like DBS. The screen design is computationally intensive but once the screen is designed, the halftoning process is again point-to-point computation process.

Halftoning algorithms assume that there is either no interaction between neighboring dots or there is additive interaction [4]. In additive interaction it is assumed that the absorptance of a halftoned patch increases linearly with the number of dots printed in the patch. But actually there is always overlapping between neighboring dots. This phenomenon is called dot gain. One or all of the following effects cause dot gain: 1) Optical dot gain caused by trapping of scattered light under the colorant, 2) Mechanical dotgain caused by spreading of colorant on the medium, and 3) Electric field caused in EP printers. The dot overlap increases as the number of dots increases. Dot clusters formed by the clustered dot screens reduces the effect of dot overlap by decreasing the number of independent dots. Clustered dot halftoning is commonly used in mass print media like newspapers, magazines, posters, etc. EP printers are unstable when printing dispersed dot halftoned image as a single dot does not develop on the paper with these printers. As a result, the clustered-dot halftoning technique is widely used with the EP printers as they give more stable dots on the paper. It is also widely used in print media. This research mainly concentrates on embedding data in the cluster dot halftoned image printed by the EP printer.

1.3 Related Work

A lot of research is going on to find techniques to detect a forged document and its source using watermarking techniques. There are two ways in which data can be embedded in an image: 1) Halftoning algorithm-level data embedding and 2) Printer mechanism-level data embedding.

Halftoning algorithm-level data embedding involves screening. Many people have worked in this area and developed some of the following techniques for embedding the data. One technique to embed information is by using stochastic screen patterns [5] and conjugate halftone screens [6] in which two screens are used to form two halftone images and the data are embedded through the correlations between two screens. Since these methods are based on pixel-by-pixel comparison, computational complex-

ity is low. However, two halftone images are required to extract the embedded data from these methods, and this makes these methods impractical. Another approach in halftoning algorithm-level data embedding is to change angular orientation of the circularly asymmetric halftone dot patterns that were written in halftone cells of digital halftoned images [7]. But producing circularly asymmetric cluster dots causes drastic visible patterns in halftoned image. Most screens grow dots in circular fashion to obtain halftoned image quality comparable to the continuous tone image. So the chances of obtaining circularly asymmeteric cluster dots are very less. This method also, is computationally complex. It is not robust and cannot embed large amounts of data without perceptual distortion. Another method in halftoning algorithm-level data embedding is to use error diffusion and search based halftoning methods to create halftone image; but this method too involves computational complexity and is very slow for real time printing applications.

Printer mechanism-level data embedding is done by regulating printer parameters at the printer mechanism-level to embed information. Two main techniques of watermarking which regulates printer parameters are: 1) Intrinsic signature and 2) Extrinsic signature [8].

Intrinsic signatures are the characteristic of a particular printer. As the printer head consists of mechanical parts, it is always possible that the two printer heads in different printers but same model may have slight variations in the manufacturing of these parts. This results in slight variations in the documents printed by these two printers. Similarly, variations exist in different models of the printers manufactured by the same manufacturer. Initial efforts were made to identify the intrinsic signatures by identifying some patterns or markings that were created on documents and images by the imperfections in the printers. Researchers at Purdue have been successful in identifying such patterns and markings in printed documents and have been able to determine the specific type of printer used i.e. inkjet or electrophotographic(EP) printer, the manufacturer of the printer and model of the printer based on features like banding, misalignment, etc. Banding [9] is a major artifact in the EP printers

and are mainly caused by fluctuations in the angular velocity of the OPC drum and results in non-uniform scan line spacing. Banding artifact appears as alternating dark and light bands perpendicular to the process direction.

A second method of watermarking is called extrinsic signature. The model of intrinsic signature, developed earlier, led to the development of a model for extrinsic signature. It was known as extrinsic signature because some artifacts or misalignments were deliberately added in the printed documents which cannot be perceived by the human eye. This was done by modulating the laser intensity in the EP printers [10]. Both of the techniques for embedding data in an image have low data-embedding capacity. It is not possible to embed large binary data into the document or image. The data to be embedded in the images and documents should contain the manufacturer's identity, printer serial number, and date on which the image or document was printed if the source of the forged document is to be identified. This additional information should be embedded on every printed page. This requires that an embedding method be such that it has large data embedding capacity. To achieve this, other methods have to be developed and implemented.

Another printer mechanism-level data embedding is using Pulse Width Modulation(PWM) in AM/FM halftoning technique [4]. However, in this research, we have restricted to developing embedding and detection technique in halftoned images obtained using clustered-dot halftoning technique.

The benefit of using printer mechanism-level data embedding method is that it cannot be easily copied or modified by anyone who wants to forge documents. So it is possible to embed the desired data in the image printed by any EP printer and is not perceptible to the unaided human eye. Only with the help of appropriate equipments and decoding techniques, one can find data embedded in an image and extract it. We use screening to obtain halftone images. A screen can be a simple cluster dot screen or hybrid screen which gives the most desirable quality of halftoned image as compared to the other type of screens. It overcomes the limitations of clustered-dot screens and dispersed-dot screens by combining both. Even though a 0 degree screen

is used, the embedding and detection method developed can be applied to screens oriented at some angle. This thesis is organized in the following manner: Chapter 2 describes the halftoning and embedding process. Two different halftoning screens are considered and the one that gives better visual quality of a halftoned image is selected. Two different embedding methods are also experimented and one that has more data embedding capacity and easy to detect is considered. Chapter 3 describes the detection process. The detection process consists of preprocessing a scanned image to remove noise, segment clustered-dots from its background, obtain lattice to determine clustered-dot cells, determine the centroid of each cluster-dot and detecting the embedded data. Chapter 4 talks about error control coding and repetition code used in this research. Chapter 5 gives the results and concludes the thesis.

2. EMBEDDING PROCESS

The objective of this thesis was to devise a method of embedding machine readable data in an image, halftoned using clustered dot halftoning technique, printed by electrophotographic(EP) printer and which can be efficiently and accurately detected from the scanned image with appropriate equipments and software. The term machine readable implies that no human observation or skill was required to detect the embedded data. Whole process of embedding and detection was made automatic, meaning that a code was developed, which reads the scanned image, preprocesses it and extracts the embedded data from the image using various image processing algorithms and techniques without human intervention. The embedded data should not be visible to human eye without aid of special equipments and should maintain the halftoned image quality obtained without the embedded data. The entire process of embedding and detection is shown in Fig. 2.1 and can be divided in two main parts - embedding stage and detection stage. Embedding stage includes halftoning technique and embedding technique which are discussed in this chapter and detection stage includes scanning, preprocessing, detecting and extracting embedded data which are discussed in the next chapter.

2.1 Embedding Process

The Embedding Process is comprised of following stages:

1) Halftoning Stage

2) Embedding Stage

Many halftoning techinques are available like clustered-dot halftoning, error diffusion, AM/FM halftoning technique, DBS, etc. This thesis is restricted to clustered-

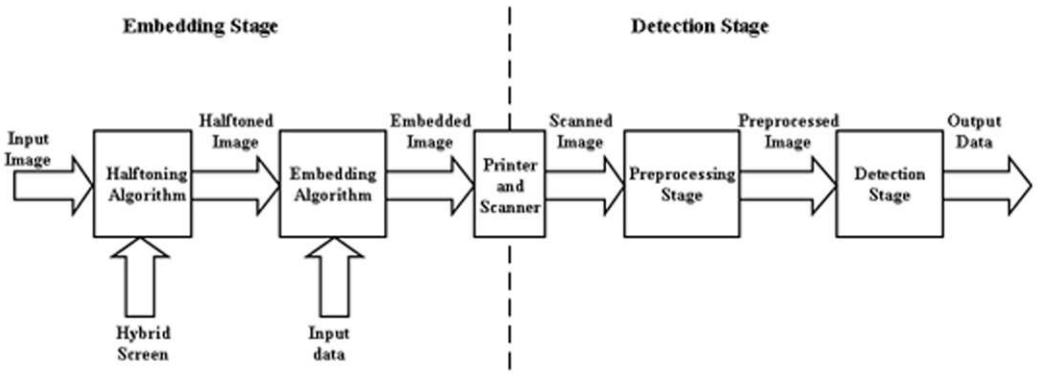


Fig. 2.1. Block diagram of embedding and detection stage

dot halftoning technique. Section 2.2 describes the halftoning process and the screen used for obtaining the halftoned image. Section 2.3 describes in detail the embedding method used in this research.

2.2 Halftoning Stage

A continuous tone image having gray level values from 0 to 255 is transformed to a halftoned image in this stage. Throughout this thesis we will refer gray levels in terms of absorptance values unless stated otherwise i.e. 0 is white and 255 is black. Halftoning is done by screening [1] to obtain clustered-dot halftoned image. Screening is a point-to-point memoryless comparison process. Fig. 2.2 shows screening process. The threshold matrix is periodically tiled over the entire continuous tone image.

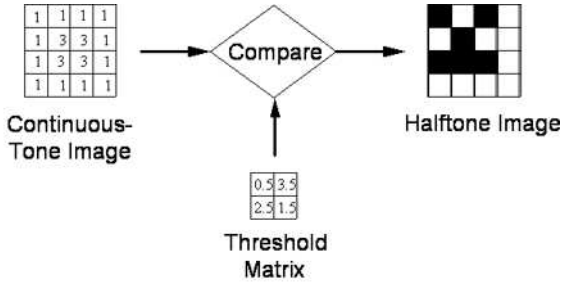


Fig. 2.2. Screening Process

2.2.1 Hybrid Screen

Hybrid Screen is based on an enhanced supercell technique for producing clustered-dot halftone image [11]. In the supercell technique, there is a set of microcells and these microcells are combined to form a single screen so as to increase the number of gray levels that can be reproduced in halftoned image. It also allows us to select any angle for the screen to get better halftoned image than with simple clustered-dot screen. It is required that the screens have rational-tangent angle but the supercell can be designed in such a way that screen as a whole have rational-tangent angle while the microcells within the screen can made to have angles close to the desired angle by making changes in the design of microcells [12]. In order to achieve more number of gray levels with supercell, there has to be an additional sequence in which microcells are selected along with microcell growing sequence. This additional sequence is known as "macrocell growing sequence". Thus the number of gray levels are increased to the product of macrocell growing sequence length and microcell growing sequence length which is same as the number of pixels in the screen. Though supercell overcomes the problem of contouring artifact due to insufficient gray levels as well as detail rendition problem due to low screen resolution, it produces texture that is noticeable to human viewer. Hybrid screen solves the texture problem in highlights while preserving the advantage of the clustered-dot screen in midtones [12]. A small region in each microcell within a hybrid screen, called the core region, does not follow the microcell growing sequence and is randomized. So the first dot placed in any microcell lies anywhere within the core region. This gives an effect of dispersed-dot screen with frequency spectrum similar to that of blue noise. It also helps in eliminating maze-like artifact. After the core region in each microcell, is completely filled with dots, the regular microcell growing sequence is followed which has same texture as that produced by simple clustered-dot screen. The order of selecting microcell is governed by the macrocell growing sequence. The microcell can be designed to have dot-hole complementary symmetry in which we also define the shadow core in hole

region and follow same design procedure for highlight core as above. Thus hybrid screen produces better quality of halftoned image as comapared to simple screen and supercell screen. Hybrid screen is also being widely used in many available printers using clustered-dot halftoning technique. Thus hybrid screen used for obtaining the halftoned image. So the developed watermarking system will be compliant with the current technology used in printers and have practical implementation.

2.2.2 Hybrid Screen Design

Hybrid screen consists of two parts - microcell and macrocell. Microcell can be viewed as simple clustered-dot screen [1] and macrocell as a group of microcells joined together. The dots in microcell grow as per microcell sequence while which microcell has to be selected is determined based on macrocell sequence. Macrocell sequence is designed using search based optimization strategies. Screen design is computationally intensive but once designed, the comparison is still point to point process. Microcells can have any shape and can be oriented at any angle. The first step in the design of hybrid screen was to decide the microcell shape and screen angle. Most basic type of hybrid screen was designed with microcell size of 4×4 pixels and screen having 0 degree angle. The size of marcocell was 4×4 microcells, 16 microcells constituted the hybrid screen.

The next step was to specify highlight and shadow core regions which gives dispersed dot texture in the halftoned image. In the core region the original microcell growing sequence is ignored and the sequence is randomized. The core shape was determined by considering the overall microcell growing sequence. As the size of microcell was 4×4 pixels, 2×2 = 4 pixels were selected for shadow and highlight core regions. And the shape of these cores were also selected to be square. After determining the core regions, the final step was to create microcell growing sequence. This growing sequence was followed only in the midtone area in hybrid screen.

Next, highlight and shadow textures were generated using DBS [13], [14]. But before that the desired number of total output levels were determined. As input image has 256 gray level values and the designed hybrid screen had total size of $(4\times4)\times16 = 256$ pixels, the number of total output levels were 256. The sequence in which pixels are turned on at each gray level is known as "dot profile function". The dot profile function was generated using constrained DBS algorithm. Finally the dot profile function for midtone levels was generated. Fig. 2.3 shows the hybrid screen used in our experiment with highlight core, midtone and shadow core areas.

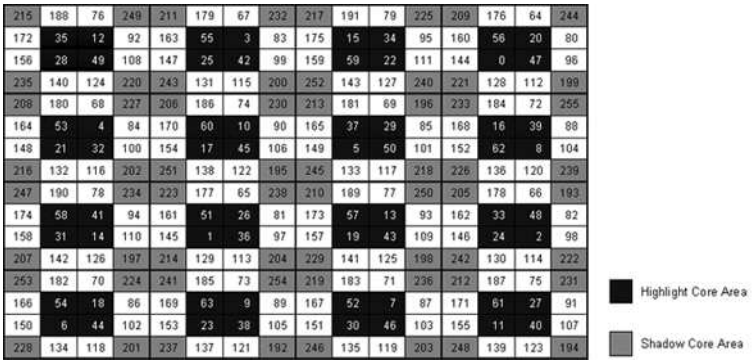


Fig. 2.3. Hybrid Screen

2.3 Embedding Stage

The halftoning stage is followed by embedding stage. The binary data can be embedded by regulating some parameters of the halftoned image in the embedding stage. These parameters can be the shape, size, orientation or location of the clustered-dots in halftoned image. The desired data(message) can be manufacturer's name, serial number of the printer, date on which it was printed etc. It is very important to ensure that the embedded message does not degrade the quality of the halftoned image or produce visual distortions in the image and it should not be percieved by the human viewer.

2.3.1 Embedding data by modulating the size of clustered-dot

Different approaches were experimented to obtain most robust method for embedding data and the one that preserves the original image quality. One method of embedding data, which was experimented, was increasing or decreasing the size of the clustered-dot. This method works in the following way. In the halftoned image, cells containing single clustered-dot were identified by determining the underlying horizontal and vertical lattices. The size of the cell should be same as screen size in case of simple screen clustered-dot halftoning and as microcell size in case of hybrid screen clustered-dot halftoning. After that the regions having constant tone were identified by scanning throughout the image. We denoted these regions as constant-tone regions and the array of cells considered in classifying these constant-tone regions was denoted as supercell. This supercell is different from that used in hybrid screen and should not be confused with it. An array of nine cells with 3 rows and 3 columns formed a supercell. Constant-tone regions were classified using a pixel by pixel comparison among all the cells of the supercell. If the size and shape of all the clustered-dots in supercell were the same then it was classified as constant-tone region. After that the size of center clustered-dot, in the supercell, was increased by one pixel for binary data bit of 1 and it was decreased by one pixel for binary data bit of 0. Which pixel in clustered-dot had to be turned on or off was decided based on the dot profile function of the screen, defined during the halftoning process. The average absorptance of the final image obtained after embedding process was same as halftoned image based on the assumption that the message bits had equal number of 1's and 0's. Fig. 2.4 shows an example of increasing and decreasing clustered-dot size by one pixel.

2.3.2 Embedding data by shifting the clustered-dots

However, the above described method had some limitations while implementing practical system. It is possible that the images may or may not have sufficient number of such constant-tone regions to embed data, thereby allowing less number of data

(a) Embedding data bit 1 (b) Embedding data bit 0.

Fig. 2.4. Embedding data by increasing and decreasing cluster-dot size.

bits that can be embedded. With hybrid screen being used in most printers, the possibility of finding nine cluster dots in neighborhood which have same shape and size becomes very difficult because the hybrid screen is designed in a way that it minimizes the repetition of dot shapes in neighborhood,which can cause contouring artifact, by randomizing the process of dot growth in core areas. The absorptance of the clustered-dots in the scanned image is dependent on the tone developed on the paper by the printer and scanner used. Thus the resultant detection method becomes dependent on devices like printer and scanner while our purpose was to develop a device independent embedding and detection method.

To overcome this drawback, another method for embedding data, which was experimented, was shifting the clustered-dots instead of increasing the size of the dots. As the screen is tiled throughout the continuous tone image in horizontal and vertical directions, clustered-dots are placed systematically forming rows and columns of the dots in halftoned image and it is possible to identify any clustered-dot shifted from its regular position. There are eight possible directions in which the clustered-dot can be shifted and it can be shifted by one or two pixels without significant merging of the dot with its neighboring dot. As the data to be embedded is binary, two out of the eight directions have to be selected for representing each symbol. The distance between the two symbols(measured by distance between centers of the shifted dots) should be as large as possible. The diagonals of the square have greater length than the four sides. So shifting the dots along the diagonal and in opposite direction gives

the maximum possible distance between the two symbols. The shifting of cluster dots was done based on the input data bits as per the following rule:

1) If the input data bit is 1 then shift the cluster dot by one pixel towards upper left corner of the microcell.

2) If the input data bit is 0 then shift the cluster dot by one pixel towards lower right corner of the microcell.

Mathematically, let $[m_1, n_1]$ be the upper left corner pixel and $[m_4, n_4]$ be the bottom right corner pixel of the cell $[m, n]$. Let H be halftoned image and E be halftoned image with data embedded in it. Let msg be the input message data bit. Then the embedding process at clustered-dot $[m, n]$ is defined as
For data bit msg = 1,

$$\begin{aligned} E[(m-1)_4, (n-1)_4] &= H[(m-1)_4, (n-1)_4] + H[m_1, n_1], \\ E[(m-1)_4, n_{1-3}] &= H[(m-1)_4, n_{1-3}] + H[m_1, n_{2-4}], \\ E[m_{1-3}, (n-1)_4] &= H[m_{1-3}, (n-1)_4] + H[m_{2-4}, n_1], \\ E[m_{1-3}, n_{1-3}] &= H[m_{2-4}, n_{2-4}] \end{aligned} \tag{2.1}$$

For data bit msg = 0,

$$\begin{aligned} E[(m+1)_1, (n+1)_1] &= H[(m+1)_1, (n+1)_1] + H[m_4, n_4], \\ E[(m+1)_1, n_{2-4}] &= H[(m+1)_1, n_{2-4}] + H[m_1, n_{1-3}], \\ E[m_{2-4}, (n+1)_1] &= H[m_{2-4}, (n+1)_1] + H[m_{1-3}, n_1], \\ E[m_{2-4}, n_{2-4}] &= H[m_{1-3}, n_{1-3}] \end{aligned} \tag{2.2}$$

To ensure accurate detection of the shifted dots and minimize false and missed detections, it was necessary to determine the range of gray level values within which the data can be embedded. Constant tone patches for various gray levels were created. These patches were halftoned using clustered-dot halftoning technique using hybrid screen and data was embedded by shifting the clustered-dots according to the rule described above. These images were printed, scanned and preprocessed using

techniques described in next chapter. Embedded data was extracted from the image. Based on the accuracy achieved with different gray levels, the range of gray level within which data should be embedded was determined. Small sized dots are unstable and may or may not develop on the paper when printed. Large sized dots tend to merge with neighboring dots due to dot gain. So the best choice was to embed data in between which corresponds to midtone area. Fig. 2.5 shows the plot of error rate with gray level values. It can be seen from the plot that the optimum range of gray level values is from 65 to 95.

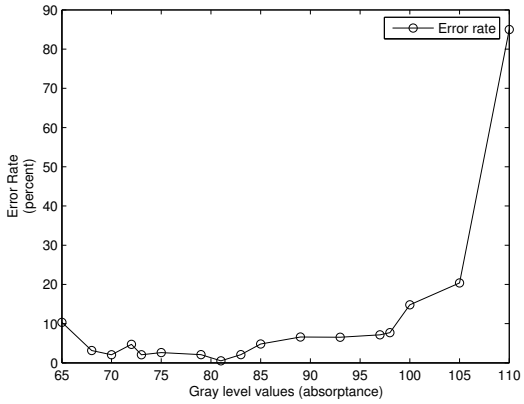


Fig. 2.5. Plot of error rate vs. gray level values

An edge in an image is marked by the sharp change in the luminous intensity. Though the luminance might be uniform on either side of an edge, it won't be uniform at the edge. The shape of the clustered-dots on the edge were random and they were not similar to any of the shapes and size of clustered-dots defined by dot profile function. If such random dots were only present in the upper or lower half or left or right half of the cell then it would appear as being shifted while detecting the shifted dots. Also if such dots were shifted for embedding data, they would end up being placed at the center of the cell rather than at the corners which was required to detect the shifted dots. As a result there were more errors in detection process when detecting near an edge. Due to this reason embedding data on the edges or in its

neighborhood was avoided and in the detection stage any shifted dots on or near an edge were ignored. The following condition was checked to detect whether an edge was present in the neighborhood of the clustered-dot being shifted.

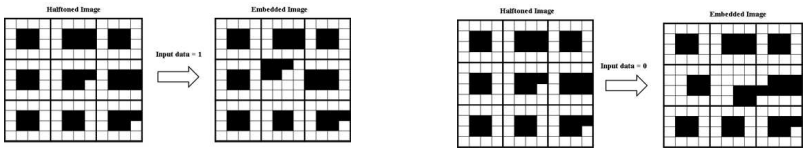
$$|A(m,n) - A(m+i,n+j)| \begin{cases} < 3\,pixels, & no\,edge \\ \geq 3\,pixels, & edge \end{cases} \quad i,j = -3,-2,\ldots,2,3 \quad (2.3)$$

.

where A = Absorptance of the cell (m,n) in terms of number of pixels turned on,
(m,n) = Clustered-dot considered for shifting

Shifting the clustered-dots diagonally caused them to merge with the neighboring dots located towards the shiftind direction while the neighboring dots located opposite to the shifting direction were separated more. When this image was printed, it tend to develop more toner on the merging dots and leave a hole where dots were separated more. Due to this, a sharp contrast in tone level was created that were visible in the printed image. To reduce this visible distortions in the image, the overlapping area of the merging dots and the size of the hole created between separated dots should be decreased. As we were shifting by one pixel, it was not possible to reduce the overlapping area of the dots but the hole size could be reduced if additional dots in the neighborhood of the shifted dot were also shifted. If a dot was shifted to upper left corner of the microcell then the dot on its right was shifted horizontally by one pixel towards left and if the dot was shifted to lower right corner of the microcell then the dot on its left was shifted horizontally by one pixel towards right. Fig. 2.6 shows an array of 3×3 clustered-dots with center clustered-dot being considered for the embedding process by shifting the dot diagonally by one pixel and its neighboring clustered-dot shifted horizontally depending on the input message data bit.

There were many other ways in which the dot pair were shifted but there was no significant improvement in the visible appearance of the image or accuracy of the detection process. After the data was embedded in the image, it was saved and printed on printer after converting it into format which bypasses the printer driver and prints image as desired. As the entire process of embedding was independent of



(a) Embedding data bit 1 (b) Embedding data bit 0

Fig. 2.6. Embedding data by shifting clustered-dot pair.

the halftoning screen used, it was not necessary to obtain the hybrid screen which produces best quality of halftoned image. Also the continuous tone image was not gamma uncorrected before producing the halftoned image as the embedding process was directly based on the information provided by halftoned image and not continuous tone image and so the results, in our experiment, won't be affected without gamma uncorrection.

3. DETECTION PROCESS

Chapter 2 described the embedding process. This chapter will describe the detection process. The entire detection process can be divided into a number of smaller processes like scanning, preprocessing, detecting, and extracting the data bits embedded in the image. The entire process of detection was made automatic such that no human skill was required to do any tasks listed above. The objective was to design a machine-readable embeddeding and detection system so that no human observation or decision was required to detect or extract the data from the image. Fig. 3.1 shows the block diagram of the detection process. The detail description of each blocks is given in the following sections.

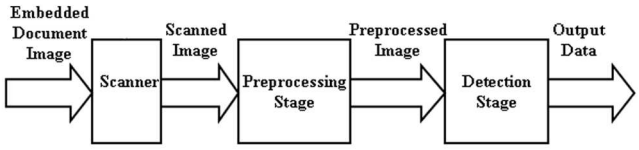


Fig. 3.1. Block diagram of detection process

3.1 Scanner

First, the printed image is scanned using a scanner. The scanner choosen must have optical resolution higher than the printer resolution on which the image was printed. Typically the printer resolution is 600dpi. Images were scanned at 2400dpi using Epson Expression XL10000 scanner in the current research.

3.2 Preprocessing

Scanned images should be preprocessed to obtain correct results from the images in detection stage as it removes the noise due to the electomechanical process in EP printers and due to scanners during the scanning process. Preprocessing involes basic image processing techniques and algorithms like tone correction, thresholding, skew angle correction, image registration and segmentation. Fig. 3.2 shows the substages of preprocessing stage.

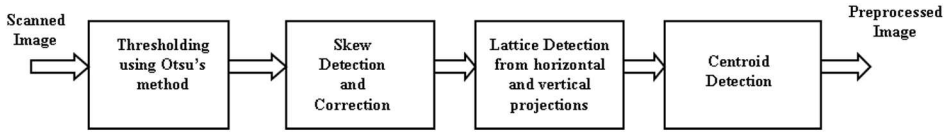


Fig. 3.2. Block diagram of preprocessing stage

3.2.1 Scanner Calibration

Calibration is the process of obtaining device independent color space values like CIE XYZ from device dependent color space like RGB. The scanner may not be properly calibrated as a result of which the gray level values obtained in the scanned image may not be the actual values. Scanner calibration [15] allows us to do tone correction of the scanned image to produce correct gray levels of the image.

3.2.2 Thresholding

Thresholding is a technique to segment objects in an image from the background on the basis of the distribution of gray levels in the image objects. The simplest is two level thresholding. The pixels with gray level greater than or equal to the threshold

are part of object and pixels with gray level less than the threshold are part of the background. Mathematically it can be expressed as:

$$x[m,n] = \begin{cases} x[m,n], & x[m,n] \geq T \\ 0, & x[m,n] < T \end{cases} \tag{3.1}$$

where,

T = Threshold

(m,n) = pixel location

$x[m,n]$ = gray level value of the pixel

Many algorithms are available to find the threshold for image segmentation. Thresholding algorithms can be parameteric or non-parameteric. Most of the algorithms use statistics of the histogram of gray level values of image to obtain threshold. The most common thresholding algorithms among researchers is Otsu's and Kittler's method and they produce better results. Otsu's method is a non parametric approach and finds optimal thresholds by maximizing the between-class variance with an exhaustive search [16]. Kittler's method optimizes a criterion function related to the average pixel classification error rate [17]. Both of these thresholding techniques were experimented with and the results obtained by both of them were similar.

3.2.3 Skew detection and correction

The next step was the skew angle detection and correction. Scanned images can be skewed during the scanning process. A very small skew is added during the printing process by a printer. This skew has to be corrected for obtaining correct results. The skew angle was determined by obtaining the slope of the top margin of the image. To achieve that, first the image was projected horizontally and vertically. The indices corresponding to the first and last non-zero values in the horizontal projection gives the top most and the bottom most margins respectively of the image. Likewise the indices corresponding to the first and last non-zero values in the vertical projection gives the left most and the right most margins respectively of the image. Now

scanning the image from the left most margin to the right most margin, the indices corresponding to the first non-zero values from the top, in each column of pixels, were determined. As the paper between two clustered-dots is white, the index corresponding to first non-zero value between two clustered-dots will give some random value which will affect the estimation of the line along the top margin of the image. To avoid this problem, for any column of pixels, if the index had value greater than the top limit, it was neglected. The top limit value was chosen as top most margin + 100 pixels. Based on the assumption that the image won't be skew drastically(more than ± 25 degrees), it was safe to take the above value as top limit. Once the set of data points(indices) for the top margin of the image were obtained, a straight line was fitted to it using least squares method(LSM). This gave us the slope of the line along the top margin of the image. Eq. 3.2 was used to calculate the slope of the top margin. Fig.3.3(a) shows the skewed image with its horizontal and vertical projections. It shows the top, bottom, left and right margins of the skewed image. Fig.3.3(b) shows the skew corrected image.

$$Slope = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n\sum_{i=1}^{n} {x_i}^2 - \left(\sum_{i=1} n x_i\right)^2} \tag{3.2}$$

where,

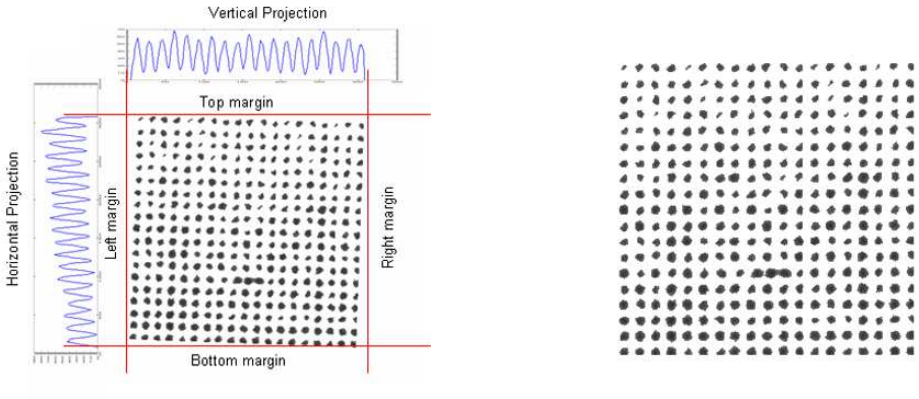
n = Number of indices corresponding to first non-zero value obtained

x_i = Value of indices corresponding to non-zero value in horizontal direction

y_i = Value of indices corresponding to non-zero value in vertical direction

The skew angle is given by $\theta = \tanh(slope)$. Skew angle obtained by this method was very accurate which is required for accurate detection of embedded data. Rotation was performed by multiplying the following rotation matrix to the image.

$$R = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \tag{3.3}$$



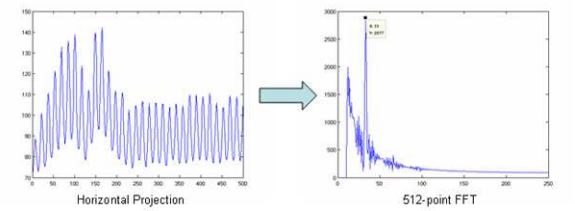
(a) Skewed image (b) Skew corrected image

Fig. 3.3. Skewed image with horizontal and vertical projections and skew corrected image

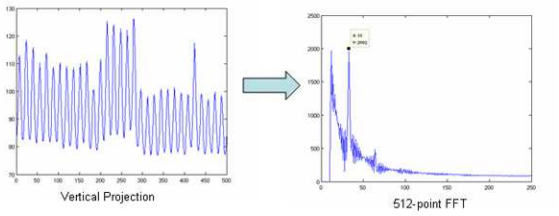
3.2.4 Lattice Detection

In clustered-dot halftoning, the screen is tiled over the entire image such that it never overlaps previously placed locations. The screen can be a square or some other shape (in case of angled screen) but we will consider square shaped screen in the following discussion. Each microcell, within the screen, is also assumed to be square shaped. Thus the clustered-dots produced are at fixed intervals along horizontal and vertical directions and are placed in a straight line. So the next step in preprocessing stage was to find underlying lattices for the clustered-dots. The size of each dot would be equal to or smaller than the size of microcell used in halftoning process. So the distance between two horizontal and vertical lattices would be equal to height and width, respectively, of the microcell times the ratio of scanner resolution to printer resolution. For example if the image is printed at 600dpi resolution by printer and scanned at 2400dpi by the scanner then each original printed pixel corresponds to $4\times4(\frac{2400}{600} = 4)$ pixels in the scanned image. So if the size of microcell used for halftoning was 4×4 pixels then the size of each dot cell in scanned image will be 16×16 pixels. The distance between two horizontal and vertical lattices was determined by calculating the frequency of occurance of the clustered-dots. This frequency was

obtained by taking horizontal and vertical projections of the image which gave a 1-D waveform. Taking N-point FFT of this 1-D waveform, the frequency f_{max} which had the highest peak value was found. But before that the 0 Hz frequency value was removed as it has highest peak value corresponding to the average value of the image. Then the period(number of pixels) of clustered-dot would be $\frac{f_{max}}{N}$. This period(number of pixels) was the distance between horizontal lattice. The process was repeated for vertical projection and the distance between vertical lattice was obtained. Fig. 3.4 shows 512-point FFT of horizontal and vertical projections of the image and the frequency corresponding to highest peak.



(a) Horizontal Projection and its FFT



(b) Vertical Projection and its FFT

Fig. 3.4. Calculating lattice length from the FFT of horizontal and vertical projections of the image

The horizontal and vertical lattice were obtained after the size of the cell had been determined. The image was projected in horizontal direction to obtain horizontal projection of the image. The projection was scanned in vertical direction from the top, to determine the index corresponding to the first non-zero value in the projection. As white colored paper was used for printing, it had absorptance value of 0 where no clustered-dots were present. Thus the index of the first non-zero value in horizontal

projection corresponds to the top most margin of the scanned image. Similarly, the index of the last non-zero value in horizontal projection corresponds to the bottom most margin of the scanned image. Dividing the difference between the top margin index and the bottom margin index by the distance between two horizontal lattices, gave us the number of cells in the image which were same as the number of clustered-dots present. One way to obtain the horizontal lattices was by recursively adding distance between two lattices, which was obtained by taking FFT of the horizontal projection, to the previous horizontal lattice starting from top margin index upto bottom margin index. But many times due to instabilities in the electromechanical process involved in printing stage, there was a possibility that some rows of clustered-dots were not placed in its expected location and were offset by a very small distance. In such cases we got many errors in detection of the embedded data. To overcome this, instead of using the static lattice detection method, we detected lattices dynamically. The horizontal and vertical projections of images gave maximas at the center of the clustered-dot and gave minimas between two clustered-dots. We denote the maximas as peak points. The indices corresponding to peak points were determined. The peaks occur at approximately fixed interval equal to the period obtained by taking FFT. The actual lattice locations were obtained by calculating the midpoint of the indices corresponding to two consecutive peaks. Thus, the lattices were separated by lattice length but at the same time it took into account the offset that could be present due to instabilities in the electromechanical process involved during printing. Fig. 3.5 shows the image with horizontal and vertical lattices. The peaks in the horizontal and vertical projections coincide with the center of the clustered-dots and the lattices lie between the two peaks.

3.2.5 Segmentation

This process was very important to remove the noise in each cell. Due to the dot gain and electromechanical process involved in printing, there was a possibility that

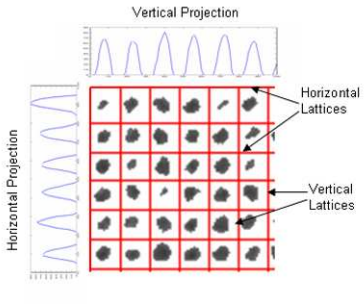


Fig. 3.5. Lattice Detection

clustered-dots might not be confined within a single cell. The clustered-dot of a given cell might get merged with clustered-dot of the neighboring cells or they might be separate but still the given cell has some portion of the neighboring cell's clustered-dot, present in it. Also during the embedding process as we shifted, the second dot in the neighborhood of the dot shifted diagonally to its right or left; it would not be confined to its own cell and part of that clustered-dot would be present in the neighboring cell of the diagonally shifted clustered-dot. Such kind of unwanted dot portions (or noise) affected the calculation for determining dot centroid which was main feature required to detect and extract the embedded data. This noise must be removed from the cell while calculating the dot centroid. Segmentation algorithm was employed individually in each cell throughout the image. If the clustered-dot of the given cell and the noise were well separated then image segmentation could be done by forming clusters of connected components and considering only the largest cluster and neglecting smaller clusters while calculating dot centroid. But in many cases it happened that the clustered-dot and noise were not well separated. In those cases it became difficult to segment the cell using above described clustering technique. So we introduced another method of segmenting the cells. The center of any cluster-dot, whether shifted or not shifted, lied within the cell and this criterion was ensured by halftoning and embedding process. The pixel at the center of clustered-dot had highest absorptance value than all other cells due to dot gain from all neighboring pixels within the cell. Even if the center pixel did not have highest absorptance one

of its neighboring pixel had highest absorptance. So the pixel with highest absorptance within a cell was always near the center of the clustered-dot with a very high probability. We determined this pixel and found the cluster of all the pixels that were connected to it and had absorptance values within certain threshold from the value of the pixel with the highest absorptance. As a result, the pixels on the outer edge of the clustered-dots were not selected and any noise which might be present in the given cell was separated from the main clustered-dot. Thus the noise was masked while calculating the dot centroid in the cell and accurate centroid calculation for the shifted dots was possible. The thresholding was done according to the following equations. For any cell X,

$$X(i,j) = X(i,j),\ if\ X(i,j) \leq max(X) - 30\ and\ X(i,j)\epsilon S \tag{3.4}$$

$$X(i,j) = 0,\ otherwise \tag{3.5}$$

where

$max(X)$ = Maximum intensity of the pixel in the cell X

$X(i,j))$ = Intensity at pixel (i,j) in cell X

S = Set of connected components to max(Y)

3.3 Detection Stage

The final stage in the detection process was the detection stage wherein the shifted dots were detected and from that embedded data was extracted. Whether the clustered-dot was shifted or not could be determined from the centoid of the dots. As most of the dots were nearly circular in shape, the centroids lied near the center of the clustered-dots. So if a dot was shifted, its centroid was also shifted from regular position and was detected by comparing with the neighboring clustered-dot centroids. So first we found the centroids for all the clustered-dots in an image.

3.3.1 Centroid Detection

The centroid of each clustered-dot was obtained using the information about the horizontal and vertical lattices obtained ealier in the preprocessing stage [15]. The horizontal and vertical lattices formed the cell within which the clustered-dot was located. The cell boundaries were defined by its upper left corner and lower right corner coordinates which were obtained by the intersection of horizontal and vertical lattices. Let the upper left corner be defined by coordinates (x_1, y_1) and lower right corner be defined by coordinates (x_M, y_N) as shown in Fig. 3.6. Then its horizontal center of mass was given by

$$C_x = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M} X(m,n)m}{\sum_{n=1}^{N}\sum_{m=1}^{M} X(m,n)}, \tag{3.6}$$

and similarly, the vertical center of mass was given by

$$C_y = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M} X(m,n)n}{\sum_{n=1}^{N}\sum_{m=1}^{M} X(m,n)}. \tag{3.7}$$

where X(m,n) = Absorptance value at pixel (m,n)

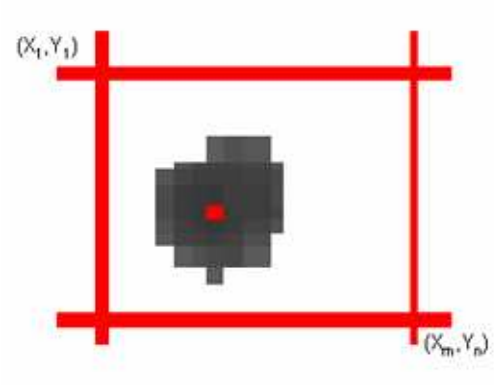


Fig. 3.6. Centroid Detection

The shifted dots were determined after the centroids were calculated. The dots were shifted diagonally so the shifted dots were determined by comparing its centroid location with the centroid of the dots on the left, right, above and below it. These

neighboring dots were not the nearest neighbors of the center dot but the neighbors of the nearest neighbors. This rule for comparing the centroids of neighboring dots was based on the following observation. Due to the electric fields in the printer, whenever a dot was shifted, it affected the neighboring dots by causing it to shift by a small amount in the direction of the shift and also as the distance between the shifted dot and its neighbors towards which the dot was shifted decreased; it caused the toner to develop in such a way that dot gain increased and dots became slightly elongated. The centroid of those neighboring dots were not placed at the actual center of the cell but slightly offset from it. While comparing a given centroid location with such neighboring dot centroid locations, there were many missed as well as false detections and the overall accuracy of the detection process decreased. To avoid this inaccuracy in detection and obtain more robust results, the dot centroid of each dot was compared with that of the dot above the nearest neighbor above and the dot below the nearest neighbor below the current dot. Similarly the dot centroid of each dot was compared with that of the dot on left of nearest neighbor on left and the dot on right of the nearest neighbor on right of the current dot. Fig. 3.7 shows the dots whose dot centroids were compared to the current dot centroid to see if the dot was shifted or not. As the image was scanned at higher resolution than printing resolution, each printed pixel now corresponded to $M = \frac{scan\ resolution}{print\ resolution}$ scanned pixels. M was denoted as magnification factor. So if the clustered-dots were shifted by one printed pixel during embedding process then the horizontal and vertical distance between the centroid of shifted dot and its neighbors was M scanned pixels. The threshold for determining shift was then taken as $\frac{M}{2}$. Once a dot was detected to be shifted diagonally, we looked for the neighboring dot that was also shifted during the embedding process to improve the visual quality of the image. This neighboring dot was shifted by one printed pixel towards left for input data bit 1 and towards right for input data bit 0. So the actual shift corresponded to M scanned pixels in scanned image. For this dot too, the threshold of $\frac{M}{2}$ was selected. The dot centroid of this dot was compared to the dot centroids of neighboring dots above and below,

similar to what was done above in case of clustered-dots shifted diagonally. After the shifted dot pair was detected, depending on which direction the dot pair was shifted, the embedded data was extracted from the image. Table 3.1 and Table 3.2 shows the values of centroid of each cluster dot. The images were printed at 600 dpi and scanned at 2400dpi so each printed pixel corresponded to 4×4 pixels in scanned image. So the diagonal or horizontal shift of one printed pixel will corresponded to 4 scanned image pixels. (Cx(i,j),Cy(i,j)) are the coordinate values of the centroid of (i,j) clustered-dot cell, where i is row number and j is column number. The Cx values of dots above and below the shifted dot and the Cy values of the dots on left and right of the shifted dot are shown in the table. Comparing the Cx and Cy values with those of shifted dot clearly shows the effect of the electromechanical process of the printer, on the 4 nearest neighbors of the shifted clustered-dot.

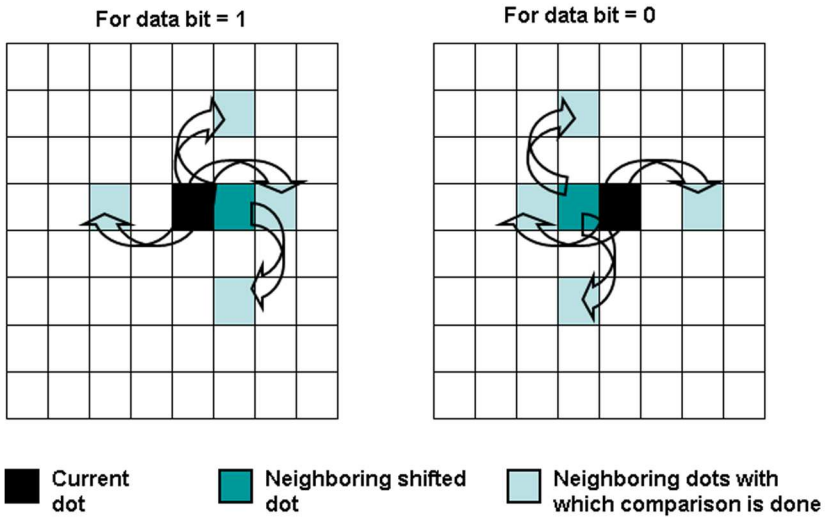


Fig. 3.7. Neighboring dots with which dot centroid of each dot and its left and right neighbors are compared to detect shifted cluster-dot.

Table 3.1
Coordinate values of the centroid of dots shifted diagonally towards upper left corner of the cell and the dots in its neighborhood

Cx(i,j)	Cy(i,j)	Cx(i-1,j)	Cx(i-2,j)	Cx(i+1,j)	Cx(i+2,j)	Cy(i,j-1)	Cy(i,j-2)	Cy(i,j+1)	Cy(i,j+2)
1880	218	1884	1885	1885	1885	220	222	221	221
6120	221	6121	6122	6121	6122	222	223	225	224
1319	235	1323	1324	1323	1324	237	240	240	240
6508	235	6512	6513	6512	6512	239	240	240	239
1511	251	1515	1516	1516	1516	254	255	254	255
5173	618	5176	5176	5176	5176	620	621	620	622
1320	794	1323	1324	1323	1324	797	797	798	798
3664	793	3667	3668	3667	3669	795	795	795	795
5383	810	5386	5389	5387	5388	812	813	813	813
935	826	937	939	938	939	829	829	829	831
4419	841	4422	4425	4422	4425	843	844	844	844
4613	841	4615	4618	4617	4618	844	844	845	844
4774	841	4777	4779	4778	4779	844	845	845	846
5158	841	5161	5164	5162	5164	844	844	845	844
5205	1017	5208	5210	5209	5210	1020	1020	1021	1022
7293	1016	7296	7297	7297	7296	1020	1021	1020	1019
4774	1033	4776	4778	4777	4778	1035	1037	1037	1038
3260	1177	3264	3265	3264	3265	1180	1181	1180	1181
5368	1177	5370	5371	5370	5371	1180	1181	1180	1181
4788	1193	4793	4794	4794	4794	1196	1197	1197	1197
4048	1208	4053	4054	4053	4055	1211	1211	1212	1212
3630	1224	3633	3634	3636	3634	1226	1228	1227	1227
5576	1225	5578	5581	5580	5580	1228	1229	1228	1228
5190	1384	5192	5193	5192	5194	1387	1388	1387	1389
5398	1400	5401	5402	5402	5402	1403	1404	1404	1405
5737	1416	5740	5740	5741	5741	1420	1421	1420	1421
4580	1560	4583	4584	4584	4584	1563	1563	1564	1565
7293	1591	7296	7298	7297	7298	1594	1594	1594	1595
5190	1608	5192	5194	5192	5194	1611	1612	1612	1611
7516	1607	7519	7520	7520	7520	1610	1611	1610	1611
5206	1943	5207	5208	5208	5208	1948	1948	1947	1948
5398	2152	5401	5401	5401	5401	2154	2156	2155	2157
4821	2168	4822	4825	4823	4825	2171	2172	2172	2172
5735	2167	5738	5738	5738	5738	2170	2171	2170	2172
7083	2167	7085	7087	7088	7088	2168	2169	2169	2169

Table 3.2
Coordinate values of the centroid of dots shifted diagonally towards lower right corner of the cell and the dots in its neighborhood

Cx(i,j)	Cy(i,j)	Cx(i-1,j)	Cx(i-2,j)	Cx(i+1,j)	Cx(i+2,j)	Cy(i,j-1)	Cy(i,j-2)	Cy(i,j+1)	Cy(i,j+2)
4443	225	4439	4439	4440	4439	220	220	221	220
4201	242	4198	4198	4198	4198	236	236	239	236
4637	257	4632	4632	4633	4633	253	252	253	252
4797	257	4793	4793	4794	4794	254	253	254	253
1504	418	1501	1500	1502	1500	414	415	415	413
4201	416	4199	4197	4197	4197	412	411	413	411
5197	417	5194	5193	5194	5193	412	413	414	413
5567	419	5565	5564	5565	5563	415	413	414	416
5744	418	5742	5741	5742	5740	415	414	415	415
5744	657	5741	5741	5743	5742	654	654	654	654
6563	659	6559	6559	6563	6560	656	655	657	655
5733	797	5742	5741	5743	5741	798	798	799	798
366	818	363	362	363	362	815	814	815	813
6371	818	6368	6369	6369	6367	814	814	815	814
6515	819	6512	6512	6513	6512	816	815	816	815
6708	849	6703	6703	6704	6704	847	846	848	845
6932	849	6928	6928	6929	6927	845	844	845	845
3285	1009	3281	3280	3281	3280	1005	1005	1006	1005
5744	1025	5742	5741	5742	5740	1021	1021	1022	1021
7317	1374	7314	7313	7315	7314	1370	1371	1371	1370
3831	1423	3829	3827	3830	3827	1419	1419	1419	1420
3848	1567	3845	3844	3844	3845	1563	1562	1564	1562
4056	1775	4053	4051	4052	4051	1771	1772	1772	1771
5791	1775	5788	5788	5789	5786	1771	1771	1772	1772
7140	1774	7138	7138	7138	7137	1770	1770	1771	1770
7668	1951	7665	7664	7664	7665	1945	1945	1947	1945
5583	1999	5578	5578	5579	5578	1997	1996	1996	1995
5759	2000	5755	5754	5757	5754	1996	1996	1996	1995
7492	2141	7487	7487	7488	7487	2137	2137	2138	2137
5791	2382	5787	5786	5788	5786	2378	2379	2379	2379
2464	2544	2461	2461	2462	2460	2541	2540	2541	2540
7284	2557	7281	7280	7282	7280	2553	2553	2554	2554
4780	2719	4776	4775	4778	4776	2715	2714	2715	2714
5790	2735	5787	5785	5786	5786	2730	2730	2730	2729
4635	2767	4633	4631	4632	4630	2763	2763	2763	2763

As the data was embedded within certain range of gray level values, the detection was also done in that range. The average absorptance of each clustered-dot was calculated based on the pixel intensity values of the constituent pixels. Eq. 3.8 was used to calculate the average absorptance of the each cell. The shifted dots having absorptance outside the predetermined range were neglected.

$$A = \frac{\sum_{n=1}^{N} \sum_{m=1}^{M} X(m,n)}{MN} \tag{3.8}$$

where A = Average absorptance of each cell

N = Number of pixels in a row of cell

M = Number of pixels in a column of cell

X(m,n) = Absorptance of the pixel (m,n) in cell

Data embedding was done in regions of uniform tone and not near edges. There were false detections whenever there was an edge. This was because at edges the dots were shaped randomly and any dot shape which was not in dot profile function could be present in the halftoned image. Whenever a randomly shaped dot has the appearance similar to the shifted dot, it leads to false detections. To avoid this, the edges were identified by comparing the absorptance in a region of 5×5 cells. If there was a drastic change in the absorptance then the probability of an edge being present in that region was high. As no data was embedded near an edge , the comparison of dot centroids at these locations was avoided.

4. ERROR CONTROL CODING

Though the method of embedding and detection described in earlier chapters gave 60% success in detection of the embedded data, we needed a system that gives 100% accurate detection. To achieve this we used error control coding. The goal of error control coding is to be able to receive and decode the transmitted data accurately without any mistakes. The concept of error control coding was developed by Shannon for any communication channel and later many codes were developed to implement it in practical system. The binary information should be conveyed without error over the channel which is susceptible to noise. The use of error control codes adds extra data bits to the information bits so that errors may be found or corrected at the receiver. Thus redundant information is added to the message data and sent over the channel.

Similar to the channel in the communication system, the printer and scanner can be seen as channel in the system described in earlier chapters. This channel is also susceptible to noise. As a result, the accuracy of the detection process was not 100%. The use of error control coding made it possible to achieve 100% accuracy in detection of the embedded data. The current chapter on error control coding is divided into two sections - one dealing with the encoding process and other dealing with decoding process.The encoding and decoding for error control coding takes place in the embedding stage and detection stage respectively as shown in Fig. 4.1.

4.1 Encoding

Many different types of error control codes [18] are available. They can be classified in two main categories:

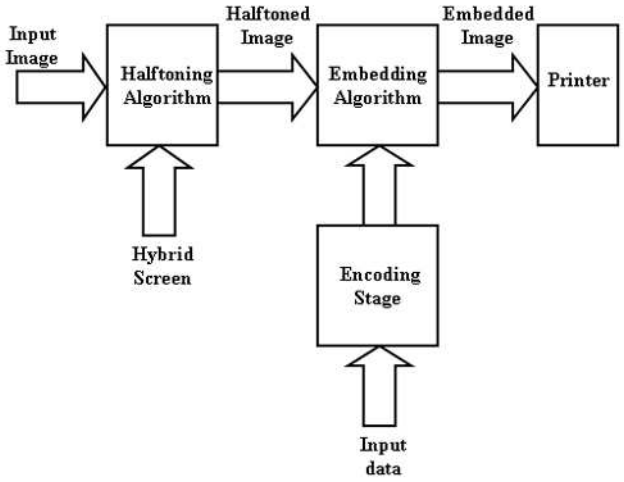


Fig. 4.1. Embedding process with error control coding

1)Based on Algebraic theory - Block codes like Hamming code, BCH, Reed and Solomon

2)Based on Probabilistic theory - Convolution codes.

In our research, we used the repetition code, which is the most simple form of error control coding. Though many robust and computationally complex codes are available, the initial efforts were to see how the system works with any type of error control coding. Repetition being a simple form of error control coding, was implemented without any computational complexity. In repetition code, each data bit is repeated k times before embedding. So if there are n message bits in the original message, encoding converts it to kn bits, which are to be embedded. Usually k is selected as odd number. The following example shows the process of encoding using repetition code.

k = 5, n = 8

Message bits : 10110010

After encoding: 1111100000111111111100000000001111100000

Length of message after encoding : 5×8=40

The data bits were embedded at random locations in the image and so there was no fixed region within which shifted dots should be detected and majority vote be taken to extract the message data. This problem was overcomed by using sync bits after each repeated k bits. Thus now there will be kn + ln = (k+l)n data bits after encoding. Here l is the number of sync bits added after each repetition. The following example shows the coding process with sync bits.

k = 5, n = 8, l = 1

Message bits : 10110010

After encoding: 11111S00000S11111S11111S00000S00000S11111S00000S

Length of message after encoding : (5+1)×8=48

where S = Sync bits

4.2 Decoding

The decoding process for repetition code is also very simple to implement. After the embedded data bits are detected from the scanned image, starting from the first bit, group the same bits between two sync bits together. Now the following decision rule is applied to each group of data bits:

Let i = Number of 1's

and j = Number of 0's

$$Databit = 1,\ if\ 3 \leq i \leq 5\ \&\ j < 3 \tag{4.1}$$

$$Databit = 0,\ if\ 3 \leq j \leq 5\ \&\ i < 3 \tag{4.2}$$

$$Databit = 11,\ if\ 8 \leq i \leq 10 \tag{4.3}$$

$$Databit = 00,\ if\ 8 \leq i \leq 10 \tag{4.4}$$

$$Databit = 10\ or\ 01,\ if\ 3 \leq i, j \leq 5 \tag{4.5}$$

Eq.4.1 and Eq.4.2 decision rules are straightforward based on the maximum occurance of any data bit. Eq.4.3 to 4.5 were to be considered in case of missing sync

bits. If the sync bits between two repetitions were not detected then the number of repetitions of 1 or 0 will be for two data bits. To avoid missing these data bits, these decision rules were decided. In Eq.4.5, whether the data bit is 01 or 10 was decided based on which one occurred first while decoding.

5. RESULTS

Two different methods for embedding data were experimented. One method was increasing or decreasing the size of the center clustered-dot in an array of 3×3 clustered-dots having the same shape and size. The other method was shifting the clustered-dot pair, selected at random locations, diagonally and horizontally. As we discussed earlier, the former method has low data embedding density as compared to the later method. Table 5.1 shows the number of data bits that can be imbedded in the image shown in Fig. 5.1 using the above embedding methods. It is clear that shifting clustered-dot allows more data to be embedded than changing the size of the clustered-dot.

Table 5.1
Data embedding capacity

	Modulating dot size	Shifting dot pairs
Number of locations where data can be embedded	85	542

Fig. 5.1 is the original continuous tone image which was used to experiment the embedding and detection techniques developed. Fig. 5.2 is the halftoned image obtained by using hybrid screen with microcell of size 4×4 pixels and macrocell of size 16×16 pixels or 4×4 microcells.

Fig. 5.3 and Fig. 5.4 shows the image after embedding data in it by shifting dot pairs by one and two pixels, respectively. Shifting dot pairs by two pixels shows drastic visual distortions in the image which are unacceptable while shifting dot pairs

Fig. 5.1. Continuous tone image

Fig. 5.2. Halftoned image (Hybrid Screen)

by one pixel has some visual distortions which are noticable only if observed with great care.

Fig. 5.3. Dot pair shifted diagonally and horizontally by 1 pixel

All the images were printed on HP5500 printer and scanned using Epson Expression XL10000 Scanner. Printer resolution was 600dpi and images were scanned at 2400dpi. Fig. 5.5 shows the locations where data was embedded. The red dots represents data bit 1 , green dots represents data bit 0 and white dots represents sync bits. Blue dots are shown to mark the hybrid screen i.e. distance between two blue dots is 4 pixels representing 4 clustered-dots. The height and width of the hybrid screen was 4 microcells each. Table 5.2 shows the the accuracy of the method in terms of bit error rate(BER).

The preliminary results show that the probability of error in the detection process is higher in the regions with high fluctuations in the gray level of the image than in a region having relatively uniform gray levels. Thus our hypothesis that embedding data should be avoided near edges is correct. Also repetition code used while embedding

Fig. 5.4. Dot pair shifted diagonally and horizontally by 2 pixels

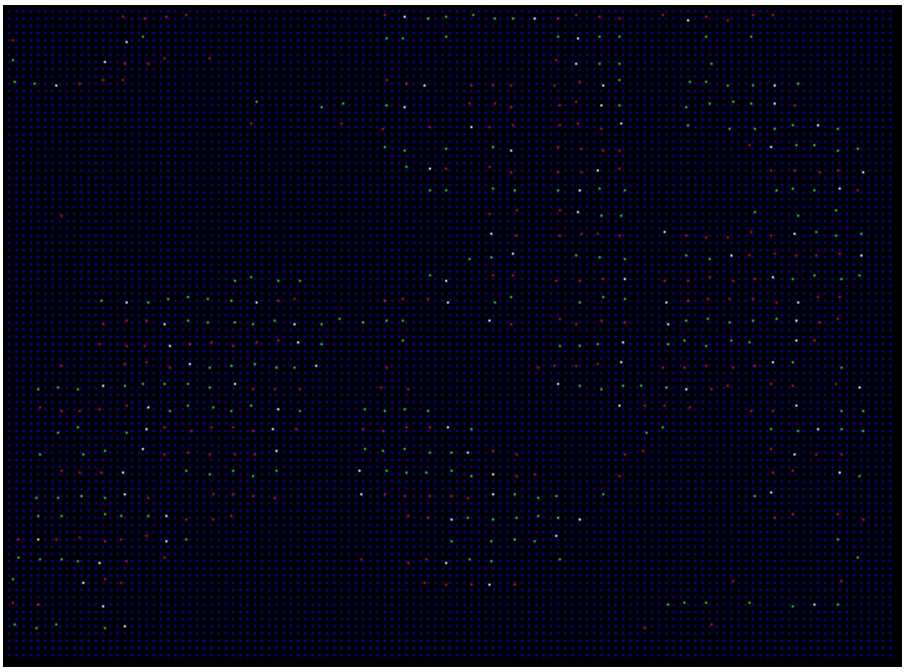


Fig. 5.5. Locations where data was embedded

Table 5.2
Bit Error Rates(BER) in two different images shown in Fig. 5.1 and Fig. 5.6. Dots pairs are shifted by one pixel

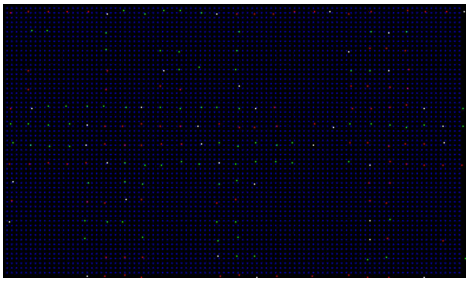
Image	Size(pixels)	Total number of bits embedded	Correctly detected bits	False Detections	Missed Detections	Original message data bits	Extracted message data bits
vase	1992×1464	547	503	27	44	91	91
window	1280×853	204	173	9	36	34	34

(a) Continuous tone image

(b) Halftoned image

Fig. 5.6. Continuous tone and halftoned images

(a) Embedded image



(b) Locations where data was embedded image

Fig. 5.7. Embedded image and locations where data was embedded

data is necessary to obtain accurate results as it is not possible to detect all embedded bits correctly beacuse of the instabilities involved in the electromechanical process in the printer.

6. IMPROVING IMAGE QUALITY USING PULSE WIDTH MODULATION

Although accuracy in detection of embedded data was achieved by the embedding method described earlier, it produced some visible distortions in the image which were not acceptable. The research was further carried on, to find an embedding method which achieved 100% accuracy in detection as well as produced no visual distortions. Earlier the clustered-dots were shifted by one pixel and yet it produced visible distortions. So we had to find a way to be able to shift the clustered-dots by half a pixel or even less such that it could not be percieved by human viewers. This was achieved by using Pulse Width Modulation(PWM) technology wherein we had the freedom to shift the clustered-dot by a fraction of the printer addressable pixel size. This chapter describes in detail the embedding and detection process using PWM and results achieved.

6.1 Embedding Process

The halftoning technique in this method was the same as described earlier. It was observed from the results obtained earlier, that the reason for visible distortions in the image with embedded data was the overlapping of the shifted clustered-dots with neighboring clustered-dots and the holes that were created at the location from which the clustered-dot was shifted. This created a sharp contrast, in a relatively uniform area, in the printed image as the overlapping dots tend to appear too dark and the pixels in its neighborhood are almost white due to the presence of the hole. To overcome this problem, the overlapping of dots should be minimized and the holes formed should be covered as much as possible.

It was possible to shift the clustered-dots by less than a pixel, using PWM, that helped in reducing the overlapping area of two clustered-dots. Also neighboring dots were shifted by some distance to cover the hole that was generated. PWM allowed the clustered-dots to be shifted by any fraction of the pixel in only horizontal direction. It was not possible to use the earlier method of shifting clustered-dots diagonally. The dots were shifted horizontally only. Shifting the dots horizontally by less than a pixel reduced the visible distortions in the image but at the same time it was very difficult to detect the shifted dots in the image. The key to reduce visible distortions was to reduce the overlap with the neighboring dots and the key to achieve accurate detection of the embedded data was to shift the dot as much as possible with respect to its neighbors with which it was to be compared. So now the clustered-dot, where the data was to be embedded, was shifted by 1.5 pixels horizontally. The two dots - one on its immediate right and one on its immediate left were shifted by 1 pixel in the same direction as the previous dot. Similarly the dots on right of the right dot and on left of the left dot were shifted by 0.5 pixel in the same direction. Thus going away on either side from the clustered-dot, where data was supposed to be embedded, the shift in the clustered-dots was gradually decreased from 1.5 pixels to 0.5 pixels. This ensured that the overlapping area of the neighboring dots was not more than 0.5 pixel and the embedded clustered-dot was shifted by 1.5 pixel with respect to its neighboring clustered-dots in the rows above and below it. If the input data bit was 1 then the array of five clustered-dots was shifted to the left and if the input data bit was 0 then the array was shifted to the right. The following equations describes the embedding process.

Let H be the halftoned image and E be the embedded image. Let (m,n) be the row and column value of the clustered-dot to be shifted and (i,j) be the pixels of any cell in H. Then, for data bit 1

$$\begin{aligned}
E_{(m,n-2)}(\frac{i+(i-1)}{2},j) &= H_{(m,n-2)}(i,j),\ \forall(i,j)\epsilon(m,n-2),\\
E_{(m,n-1)}(i-1,j) &= H_{(m,n-1)}(i,j),\ \forall(i,j)\epsilon(m,n-1),\\
E_{(m,n)}(\frac{i+(i-2)}{2},j) &= H_{(m,n)}(i,j),\ \forall(i,j)\epsilon(m,n),\\
E_{(m,n+1)}(i-1,j) &= H_{(m,n+1)}(i,j),\ \forall(i,j)\epsilon(m,n+1),\\
E_{(m,n+2)}(\frac{i+(i-1)}{2},j) &= H_{(m,n+2)}(i,j),\ \forall(i,j)\epsilon(m,n+2). \qquad (6.1)
\end{aligned}$$

And for data bit 0

$$\begin{aligned}
E_{(m,n-2)}(\frac{i+(i+1)}{2},j) &= H_{(m,n-2)}(i,j),\ \forall(i,j)\epsilon(m,n-2),\\
E_{(m,n-1)}(i+1,j) &= H_{(m,n-1)}(i,j),\ \forall(i,j)\epsilon(m,n-1),\\
E_{(m,n)}(\frac{i+(i+2)}{2},j) &= H_{(m,n)}(i,j),\ \forall(i,j)\epsilon(m,n),\\
E_{(m,n+1)}(i+1,j) &= H_{(m,n+1)}(i,j),\ \forall(i,j)\epsilon(m,n+1),\\
E_{(m,n+2)}(\frac{i+(i+1)}{2},j) &= H_{(m,n+2)}(i,j),\ \forall(i,j)\epsilon(m,n+2). \qquad (6.2)
\end{aligned}$$

The embedding of the data near edges in an image was avoided as in the earlier method. To improve accuracy of the detected data, the repetition code was used as described in earlier method.

6.2 Detection Process

The preprocessing comprised of various stages as before. The thresholding, skew detection and correction, and lattice detection methods were the same as described in Section 3.2. The detection stage in this method is different than the earlier method. Here, the method of detecting centroid had slight variation. Now the centroid detection consisted of two parts - precentroid detection and centroid detection. The segmentation was included in the precentroid detection and so there was no need to

have separate segmentation stage. The precentroid detection and centroid detection methods are described below.

6.2.1 Precentroid Detection and Centroid Detection

As the clustered-dots were now shifted by more than a pixel, they lay on the vertical lattices, which were detected in lattice detection stage, causing nearly half of the clustered-dot to be in one cell and the other half in the neighboring cell. The earlier used centroid detection method was not able to correctly detect the centroids of the shifted dots. So now in the precentroid stage, centroids of only those dots, which were shifted, were detected. This was done by scanning each row of clustered dots from left to right and going from top to bottom. At each vertical lattice in a given row, it was checked if any portion of the clustered-dot overlapped on the lattice or not. If no portion of the dots, on either side of the vertical lattice, overlapped the lattice then no action was taken. Otherwise the leftmost and the rightmost edges of the overlapping clustered-dot were found. This was done by segmenting the two cells on either side of the vertical lattice using connected components technique. If the width of the clustered-dot was less than or equal to the horizontal length of the cell then its centroid was calculated using Eq. 3.6 and 3.7 in the region between leftmost and rightmost edge. No action was taken if the width was greater than the horizontal length of the cell. Whether the clustered-dot was shifted from left to right or from right to left was decided based on which side had greater portion of the clustered-dot. If the portion of the dot on the left of the vertical lattice was more than on the right then the dot was shifted from left to right. Similar decision was taken for dots shifted from right to left. Fig. 6.1 shows a shifted clustered-dot with lying on vertical lattice and having variance less than the cell length. The center red dot is the detected centroid.

After the centroids of shifted dots were detected, the centroids for remaining clustered-dots were calculated using the same technique as explained in section 3.3.1.

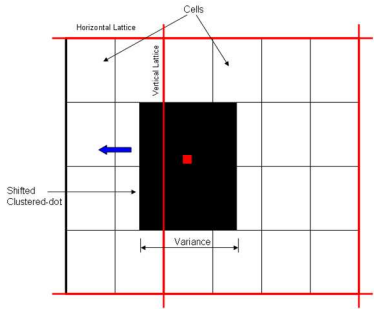


Fig. 6.1. Centroid detection of shifted dot in precentroid detection stage

To ensure better detection of the embedded data, while detecting the centroids of the clustered-dots in the neighborhood of the shifted dots detected in precentroid stage, care was taken to remove the portion of the shifted dot that would be present in the current cell.

The actual data is extracted from the data bits detected using the decoding technique described in section 4.2.

6.3 Results

The above described method was implemented in a constant tone image having gray level value of 55. The image size was 1500×1500 pixels. 80 message bits were embedded in the image after encoding using repetition code with k repetitions and adding sync bits after each repetition. The detected BER for this image was 0 and all 80 message data bits were correctly detected. Fig. 6.3 shows the embedded image. The embedded data is not perceptible at all.

This embedding method was then implemented in one of the images. Fig. 6.3 shows the embedded image. The embedded data is not perceptible in this image too. 25 message data bits were embedded in the image after encoding them using repetition code. 22 data bits were detected correctly and 3 data bits were missed causing the BER of 12%. Thus to achieve higher accuracy and decrease the BER, more accurate detection methods can be developed. Other possibility is to use error

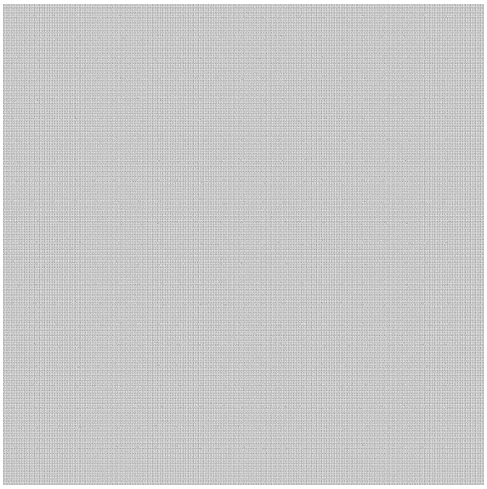


Fig. 6.2. Embedded constant tone image using PWM technique

control coding techniques like BCH, etc. in addition to the repetition code to achieve higher accuracy. But the density of data bits embedded will decrease with more redundancy added.

Fig. 6.3. Embedded image using PWM technique

7. CONCLUSION

7.1 Conclusion

We described a method of embedding machine-readable data in the hardcopy image halftoned using the clustered-dot halftoning technique and printed on EP printers. The halftoned image was obtained using hybrid screen and it was shown that it produces better halftone image quality than that produced by using simple screen. Different embedding methods were described and the method that achieved maximum embedding capacity was implemented. In this method, clustered-dots were shifted diagonally for embedding data bits. The maximum distance by which we can shift the dot without producing drastic visible distortions in an image was one pixel. Also to reduce the visible distortions in the image, the neighboring dot was shifted horizontally to fill in the hole that was created by shifting the dot diagonally. Thus a dot pair was shifted in embedding process. Error control coding was used to ensure accurate detection of data bits. Repetition code was used with each bit repeated five times. We also described a method of detecting the embedded data from the scanned image. The image was scanned and preprocessed to remove the noise from the image. The detection was done by determining the centroid of each dot and checking if the centroid was shifted or not with respect to neighboring dot centroids. The 8 nearest neighbor's dot centroids were not considered while comparing the dot centroids because they tend to be displaced from their normal position due to electromechanical process in printers and merging of dots with shifted dot due to dot gain and toner development phenomenon. This effect was shown in Table 3.1 and Table 3.2 with dot centroid values of shifted dots and dots in its neighborhood. The message data bits

were extracted from the detected bits by following decoding algorithm developed. The correct detection rate was 80-90% and with error control coding message was decoded with 100% accuracy. Thus an entire system for embedding and detecting machine-readable data in images, halftoned using clustered dot halftoning technique and printed on EP printers, was developed. This printer mechanism-level watermarking method is robust to noise and is difficult to be modified by any person trying to forge images.

Although the above given method provided accurate results in detecting the embedded data, there were some visible distortions in the image due to shifting of the clustered-dots. Efforts were made to overcome this drawback of the earlier method by using PWM technology used in EP printers. PWM allowed shifting of clustered-dots by a fraction of pixel length in horizontal direction. The dots were now shifted horizontally instead of diagonally for embedding the data. The visible distortions in the image were reduced by reducing the overlapping area for the clustered-dots in the neighborhood of the shifted dots by shifting the neighboring dots by some distance. Thus the entire neighborhood was altered while embedding data. These shifted dots were successfully detected in the constant tone images and were detected with good accuracy in normal images. These preliminary results show that PWM should be used to reduce visible distortions in the image while efforts can be made to improve the detection process. Many factors are involved in the development of toner at any particular location on paper and these factors being random processes, it is difficult to control them. So repetition code should be used while embedding the data but it is not sufficient for error detection and correction. Other error control coding techniques like BCH, Reed-Solomon, etc. should be considered to achieve robust detection of the embedded data.

LIST OF REFERENCES

LIST OF REFERENCES

[1] R. Ulichney, *Digital Halftoning*. Cambridge, MA: The MIT Press, 1987.

[2] F. Baqai, J. H. Lee, U. A. Agar, and J. P. Allebach, "Digital color halftoning," *IEEE Signal Processing Magazine Special Issue on Color Image Processing*, vol. 22, pp. 87–96, January 2005.

[3] J. P. Allebach, "Selected papers on digital halftoning," *Bellingham, WA, SPIE Milestone Series*, 1999.

[4] S. Suh, J. P. Allebach, G. T.-C. Chiu, and E. J. Delp, "Printer mechanism-level data hiding for halftone documents," in *Proceedings of the IS&T's NIP22: International Conference on Digital Printing Technologies, Denver, CO*, pp. 436–440, September 2006.

[5] K. T. Knox, "Digital watermarking using stochastic screen patterns." U.S. Patent 5734752.

[6] S. G. Wang, "Digital watermarking using conjugate halftone screens." U.S. Patent 5790703.

[7] R. T. Tow, "Methods and means for embedding machine readable digital data in halftone images." U.S. Patent 5315098.

[8] G. N. Ali, P.-J. Chiang, A. K. Mikkilineni, J. P. Allebach, G. T.-C. Chiu, and E. J. Delp, "Intrinsic and extrinsic signatures for information hiding and secure printing with electrophotographic devices," in *Proceedings of the IS & T's NIP19: International Conference on Digital Printing Technologies, Volume 19, New Orleans, LA*, pp. 511–515, September 2003.

[9] A. K. Mikkilineni, G. N. Ali, P.-J. Chiang, G. T.-C. Chiu, J. P. Allebach, and E. J. Delp, "Signature-embedding in printed documents for security and forensic applications," in *Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents VI,Volume 5306, San Jose, CA*, pp. 455–466, January 2004.

[10] P.-J. Chiang, G. N. Ali, A. K. Mikkilineni, E. J. Delp, J. P. Allebach, and G. T.-C. Chiu, "Extrinsic signatures embedding and detection for information hiding and secure printing in electrophotography," in *Proceedings of the 2006 American Control Conference,Minneapolis, Minnesota*, pp. 2539–2544, June 14-16 2006.

[11] G.-Y.Lin and J.P.Allebach, "Generating stochastic dispersed and periodic clustered textures using a composite hybrid screen," *IEEE Transactions on Image Processing*, vol. 15, no. 12, December 2006.

[12] C. H. Lee and J. P. Allebach, "The hybrid screen: Improving the breed." draft - to be submitted to IEEE Trans. on Image Processing, 2007.

[13] D. Kacker, T. Camis, and J. P. Allebach, "Electrophotographic process embedded in direct binary search," *IEEE Transactions on Image Processing,Vol. 11, No. 3*, March 2002.

[14] F. A. Baqai and J. P. Allebach, "Halftoning via direct binary search using analytical and stochastic printer models," *IEEE Transactions on Image Processing,Vol. 12, No. 1*, March 2003.

[15] E. Bernal, J. P. Allebach, and Z. Pizlo, "Improved pen alignment for bidirectional printing," *Journal of Imaging Science and Technology 51(1)*, Feb 2007.

[16] P.-S. Liao, T.-S. Chen, and P.-C. Chung, "A fast algorithm for multilevel thresholding," *Journal of Information Science and Engineering*, 2001.

[17] J. Kittler and J. Illingworth, "Minimum error thresholding," *Pattern Recognition Society, Pattern Recognition*, vol. 19, no. 1, pp. 41–47, 1986.

[18] S. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice Hall, 1995.